Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation
CRT-2005-09

TOWARDS A GUIDED COOPERATIVE SEARCH

by

Alexandre Le Bouthillier
Teodor Gabriel Crainic
Peter Kropf

April 2005

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone : (514) 343-7575 / Télécopieur : (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca

HEC ÉCOLE POLYTECHNIQUE MONTRÉAL Université de Montréal

105-03980/

PUBLICATION
CRT-2005-09

TOWARDS A GUIDED COOPERATIVE SEARCH

by

Alexandre Le Bouthillier[1]
Teodor Gabriel Crainic[2]
Peter Kropf[1]

The research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

[1] Département d'informatique et recherche opérationnelle and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (alexleb@crt.umontreal.ca, kropf@iro.umontreal.ca)

[2] Département de management et technologie, Université du Québec à Montréal, and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (theo@crt.umontreal.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2005

Abstract

We present a framework for a *guided parallel cooperative search* for combinatorial optimization based on the *central memory* multi-thread cooperative search concept. The proposed mechanism endows cooperative search with capabilities to create new information and guide the global search. Based on patterns shared by several of the solutions exchanged among the search threads, information is sent to individual meta-heuristics about promising and unpromising patterns of the solution space. This process results in a better coordination between the individual methods and the possibility to guide the diversification and intensification of the global search. We apply this method to the Vehicle Routing Problem with Time Windows. Experimental results on an extended set of benchmark problem sets illustrate the benefits of the proposed methodology.

Keywords: parallel computation, parallel cooperative search, Vehicle Routing Problem with Time Windows

Résumé

Nous présentons le concept de *recherche parallèle coopérative dirigée* pour l'optimisation combinatoire, basé sur l'approche de recherche méta-heuristique coopérative avec *mémoire centrale*. Ce concept enrichit la recherche coopérative par un mécanisme de création de nouvelle information et la possibilité de diriger la recherche globale. La nouvelle information est générée à partir de patrons communs à plusieurs des solutions échangées et permet d'identifier des régions de l'espace des solutions qui paraîssent attirantes ou, au contraire, non désirables. L'information est ensuite envoyée aux méta-heuristiques individuelles afin de les guider dans leurs recherches. La coordination des recherches individuelles est ainsi améliorée. De plus, il est maintenant possible de guider la recherche globale vers des phases d'intensification et de diversification. Nous présentons une application de ce concept au problème de tournées de véhicules avec fenêtres de temps. Les résultats obtenus pour un ensemble étendu de problèmes tests démontrent les avantages de la méthodologie proposée.

Mots clés : calcul parallèle, recherche parallèle coopérative, problème de tournées de véhicules avec fenêtres de temps

1 Introduction

Efficiency and robustness are the primary qualities of good meta-heuristics and parallel computing may significantly enhance these characteristics. Three forms of parallelism can be applied to meta-heuristics: (i) Division of computer-intensive tasks at a low algorithmic level, (ii) Explicit domain decomposition of the solution or search space, (iii) Multi-thread search. An independent multi-thread search produces the best solution among those found by each independent method. Multi-thread cooperative search implements a mechanism that allows information (e.g., solutions) to be exchanged among the search threads. Several such mechanisms have been proposed in the literature and in many cases cooperative multi-search offered superior performance in terms of better solutions and shorter resolution times (Crainic and Toulouse 2003).

While all parallelization strategies may speed up the resolution, cooperative search methods may also increase the robustness of the global search (Crainic and Toulouse 2003). Cooperative search combines the efforts of several independent meta-heuristics by using a so-called solution warehouse (according to the information stored, the names "adaptive" and "central memory" are also used). This device receives "good" solutions from the search threads and, on demand and according to their own internal logic, provides them in return with solutions to, for example, diversify the search. This simple mechanism allows the asynchronous communication and exchange of solutions that influences the search trajectory of each method. Enhanced with simple extraction rules for the returned solutions, simple cooperation was successfully used to address a number of difficult combinatorial problems: network design (Crainic and Geandreau 2002, multi-commodity location-allocation (Crainic, Toulouse and Gendreau 1995), circuit partitioning (Aiex *et al.* 1998) and Vehicle Routing Problem with Time Windows (VRPTW) (Le Bouthillier and Crainic 2005).

From a conceptual point of view, the solution warehouse contains useful indirect information for the global search. Thus, for example, the history of solution discovery and the frequency of appearance of certain attributes in solutions of particular quality can be used to create new information about the search space. This process of information creation can transform the solution warehouse in an intelligent data warehouse that holds more complex information and guides individual methods towards promising or unexplored regions.

The goal of this paper is to present a mechanism that endows cooperative search with capabilities to create new information and guide the global search. The proposed identification pattern mechanism sends information to individual meta-heuristics about promising and unpromising patterns of the solution space. By fixing or prohibiting specific solution attribute values in particular search methods, we can focus the search to desired regions. This mechanism may thus be applied to enforce a better coordination between the individual methods and control the diversification and intensification of the

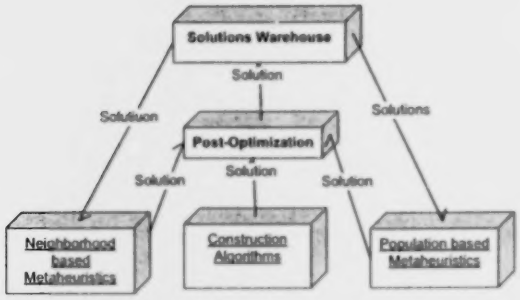

Figure 1: Abstract Framework of a Cooperative Search

global search. We apply this mechanism to the Vehicle Routing Problem with Time Windows (VRPTW). Experimental results on an extended set of benchmark problem sets illustrate the benefits of the proposed methodology.

The main contribution of this paper is the introduction of an enhanced cooperative search mechanism that creates new information from exchanged solutions and perform global intensification and diversification phases. The proposed framework does not suppose any specific problem structure and may thus be applied to a wide range of combinatorial problems. The paper also reports very good solutions on benchmark problem sets for the VRPTW: 139 best known results found and a new best known average on the problems of size 200 and 1000.

The paper is organized as follow. Section 2 briefly presents a framework of cooperative search. Section 3 introduces the pattern identification mechanism and the global search phases. Section 4 details the implementation of a guided cooperative search applied to the VRPTW. Section 5 presents the computational results and analyzes them both from the point of view of solving the VRPTW and from that of the performance of the parallel strategy. Conclusions and perspectives are the subject of the last section.

2 A Cooperative Search Framework

Figure 1 illustrates the cooperative search framework made up of a number of independent processes (threads) of possibly different types, which communicate through the solution warehouse. A search tread either heuristically constructs new solutions, or executes a neighborhood-based improving meta-heuristic through the search space, or implements a population-based meta-heuristic (e.g., evolutionary algorithms, Scatter Search, Path relinking), or performs post-optimization procedures (e.g., intensive local search) on solutions in the solution warehouse. Improving meta-heuristics, such as Tabu search, aggressively explore the search space, while population-based methods contribute toward

increasing the diversity of solutions exchanged among the cooperating methods. When the same meta-heuristic is used by several search threads, the initial solution and particular setting of a number of important search parameters differentiate each search thread from the others.

The cooperation aspect of the parallelization scheme is achieved through asynchronous exchanges of information. Information is shared through a solution warehouse or pool of solutions. In this scheme, whenever a thread desires to send out information (e.g., when a new local optimum is identified), it sends it to the pool. Similarly, when a thread accesses outside information (to diversify the search, for example), it reaches out and takes it from the pool. Communications are initiated exclusively by the individual threads, irrespective of their role as senders or receivers of information. No broadcasting is taking place and there is no need for complex mechanisms to select the threads that will receive or send information and to control the cooperation. The solution warehouse is thus an efficient implementation device that allows for a strict asynchronous mode of exchange, with no predetermined connection pattern, where no process is interrupted by another for communication purposes, but where any thread may access at all times the data previously sent out by any other search thread.

The solution warehouse keeps the information in an order appropriate for the exchange mechanism considered. To fully characterize the cooperation process, one has to specify (i) the information which is to be shared; (ii) the particular methods that makeup the cooperative search; (iii) when and how communications occur; (iv) the utilization each thread makes of the imported information (Crainic and Toulouse 2003). The information exchanged among cooperating procedures has to be meaningful, in the sense that it has to be useful for the decision process of the receiving threads. Information indicative of the current status of the global search or, at least, of some individual meta-heuristic is, in this sense, meaningful.

Two main classes of cooperation mechanisms are found in the literature, based on partial and complete solutions, respectively. Adaptive memory methods (Rochat and Taillard 1995) store partial elements of good solutions and combine them to create new complete solutions that are improved then by the cooperating threads. Central memory approaches exchange complete elite solutions among neighborhood and population-based meta-heuristics (Crainic, Toulouse, Gendreau 1995, Crainic and Toulouse 2003, Le Bouthillier and Crainic 2005).

In a simple central memory cooperation scheme (e.g., Le Bouthillier and Crainic 2005), threads share information about their respective good solutions identified so far. When a thread improves the imported solution or when it identifies a new best solution, it sends it to the solution warehouse. This scheme is intuitive and simple, and it satisfies the meaningfulness requirement. The selection of the methods involved in cooperative search should be oriented toward obtaining: (i) Good quality solutions; (ii) A broad diversity

of solutions to facilitate the discovery of promising regions; (iii) The rapid production of intermediate solutions to feed the information exchange mechanisms; (iv) A mechanism that combine various solutions to create diversity; (v) A mechanism that has the ability to escape local optima.

The solution warehouse is thud the core of the cooperation mechanism. It keeps good solutions and is dynamically updated by the independent search processes. The pool of solutions forms an elite population from which the independent procedures require solutions at various stages of execution. Solutions are ordered according to a predetermined utility measure that quantifies the solution quality. The solution utility can be its objective value or a combined measure of solution's properties.

Independent methods send their improved solutions to the post-optimization algorithms. Theses solutions are considered in-training until they have been post-optimized and sent as Adult solutions to the solution warehouse. Duplicate solutions received in the solution warehouse are eliminated.

All requests for solutions initiated by the independent processes are sent to the solution warehouse that responds by sending an Adult solution. Solutions are selected randomly according to probabilities biased toward the best based on the same function used to order solutions in the solution warehouse.

The population size in the solution warehouse is set relatively to the problem size and the worst results are eliminated as needed. No direct communications take place between processes thus enforcing their independence and the asynchronous mode of exchange. This scheme makes the cooperation design simpler and, eventually, allows easy modification of the parallel system by adding new methods or dropping inefficient ones. Moreover, it does not assume any specific problem, which makes it equally relevant for problems where solution components may be easily defined (e.g., the routes in vehicle routing problems) and for problems where such structures are much less apparent (e.g., network design). The goal now is to improve upon this simple cooperating scheme by extracting new knowledge from the information exchanged, to yield a more efficient global search.

3 Pattern Identification Mechanism and Global Search

In this section, we introduce a mechanism to extract knowledge from the information exchanged and guide each search method towards promising or unexplored regions of the solution space. It uses a pattern identification mechanism on the solutions present in the solution warehouse that is then used to fix or prohibit specific solution attributes (e.g., arcs in network-based problems) for part of the search performed by particular

individual meta-heuristics. One may therefore constrain the search space of particular meta-heuristics in cooperation and thus perform global intensification and diversification phases to guide the exploration of the solution space and control the quality and diversity of the solution warehouse population.

To enhance the clarity of the presentation, we focus in the following on problems that may be described in terms on inclusion/exclusion of arcs in given networks. Network design, traveling salesman, and vehicle routing problems are important combinatorial problem classes that belong to this category. We emphasize, however, that the concepts presented in this section may be extended to any problem definition and solution attribute.

3.1 Pattern definition

For the class of problems described above, a very simple and general pattern definition may be based on the inclusion of arcs in particular solutions.

Consider the frequency of inclusion of arcs in a given subset of the solution warehouse. In particular, this subset may be the entire population, an elite (e.g., with solution in the 10% best), average (between the 10% and 90% best) or worst (the last 10%) group of solutions. An arc with a high frequency in a given group signals that the meta-heuristics participating to the cooperation have often produced solutions that include that arc. Tagging solutions to identify the last algorithm that sent it, one may induce similar information by subset of participating algorithms as well. When the frequency of inclusion of several arcs is considered, patterns emerge among the solutions of the solution warehouse or the specific group examined.

We define a pattern of length n as a subset of arcs of cardinality $n = 1, 2, \ldots,$ maximum number of arcs in the problem definition. A frequent (infrequent) pattern relative to a set of arcs is built of arcs with high (low) frequency of appearance in the solutions of the set. High (low) frequency arcs are selected sequentially in decreasing from the highest (increasing from the lowest) frequency value.

We may select patterns from specific subpopulations (e.g., elite, average, and worst) and compare the rate of appearance of a specific pattern between them. These comparisons form the basis of the guidance mechanism proposed in this paper.

3.2 Comparing pattern-appearance frequencies among subpopulations

We define an *in-pattern* as a pattern that actually appears in at least one solution of the subset of solutions considered, as opposed to a *statistical pattern* that does not appear in any solution of the subset. A statistical pattern is thus only a consequence of the statistical process of accounting for the frequency of individual arcs

Consider an *in-pattern* of length n common to the three sets of elite, average, and worst solution groups. Two meaningful situations can occur with respect to the frequency of appearance of the pattern in these sets as one move from the elite to the average to the worst subpopulation: The frequency is either increasing or it is stable or decreasing. We call the *in-pattern unpromising* in the first case and *promising* in the latter.

Promising and unpromising patterns may then be used to constrain for a certain time the search space of particular methods in the cooperation and thus induce global intensification and diversification phases, as described in the next subsection.

3.3 Using pattern identification - Global search

Global search intensification and diversification phases may be triggered by fixing (including) or prohibiting (excluding) arcs in the solutions a given meta-heuristic explores during a certain period.

Consider an unpromising pattern of length n. To intensify the search around solutions with "good" attributes, one prohibits the arcs defining the pattern. On the other hand, fixing these arcs for a number of iterations will diversify the search relative to the current set of "good" attribute values. Symmetrically, given a promising pattern of length n, one intensifies the search by fixing the arcs in the pattern, while prohibiting them diversifies it.

We define the global search as the cumulative search effort of the individual methods. To prevent individual methods from converging too rapidly, we favor diversification by prohibiting arcs during the initial phases of the global search. Later on, we encourage intensification by fixing promising arcs to enforce the exploration of theses promising regions of the solution space. We also vary the length of the patterns as a mean of modulating the intensity of global diversification and intensification phases and thus influence the evolution of the diversity and quality of solutions in the solution warehouse.

At the beginning of the search there is not sufficient information gathered as the solution space is insufficiently explored. Therefore, the solution warehouse cannot be

Figure 2: Guided Cooperative Search applied to VRPTW

considered representative of the search of the individual meta-heuristic, even when the population is diverse. Consequently, initially, we build patterns of increasing length from the average subpopulation only, in order to identify more rapidly promising ones. As the search progresses, patterns from elite and worst subpopulations are built as described above. When a statistical pattern is found, we reduce the length of the pattern until an in-pattern is obtained.

Several modifications must be made to the initial framework presented in the previous section. The solution warehouse now includes a process to identify and manage patterns. This process includes decisions on when to compute particular patterns and to what individual meta-heuristic to send them. It becomes, in fact, the "new" global meta-heuristic corresponding to the global guided cooperative search. As for the solution warehouse, it now contains solutions and pattern information, thus a data warehouse. Communications from the solution warehouse to the individual meta-heuristics are modified as well. The appropriate pattern and instructions on fixing or prohibiting arcs are sent along with the solution (selected according to the original criteria). With respect to the individual meta-heuristics, each needs to be modified to cope with the instructions relative to fixing or prohibiting arcs. Figure 2 illustrates the guided cooperative search framework as applied to the VRPTW.

4 Guided Cooperative Search for the VRPTW

To illustrate the mechanism described in the previous sections, we developed a guided cooperative search for the Vehicle Routing Problem with Time Windows (VRPTW). The application is described in this section. Experimental results are discussed in the next section.

The VRPTW is a well-known combinatorial problem that has been extensively studied

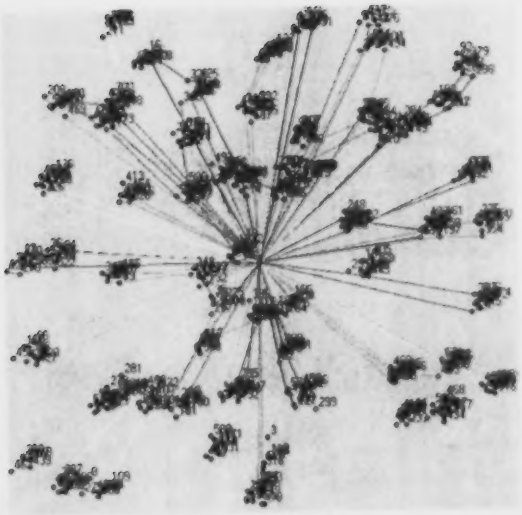

Figure 3: Best known solution for problem C1-6-4

and is thus well suited for benchmarking. We address the single depot VRPTW, as illustrated in Figure 3, where one is given a set of customers with known positive demands and specific time intervals when service can be provided. A fleet of homogeneous vehicles of known capacity is available at a given depot to perform this service. The objective is to find a set of closed routes (or tours) that start and end at the depot within its opening hours, such that the total cost of performing the service is minimized, customers are visited and served during their specified time windows, and vehicles are not overloaded.

In the problem version we address, cost is a combination of two factors: the number of vehicles (routes) used and the total distance traveled. A high cost is associated with vehicle utilization to enforce the search towards solutions with reduced number of vehicles. Each customer is visited only once. A vehicle cannot arrive later than the customer's closing time, but is allowed to arrive before the associated opening time, in which case it waits, without explicit penalty, until the customer is ready. Once the service starts, it is carried on until completion, even if the service ending time might be later than the expiration of the time window. Cordeau et al. (2002) review problem variants, formulations, and solution methods for the VRPTW.

Le Bouthillier and Crainic (2005) presented a cooperative parallel method for the VRPTW based on the simple solution warehouse mechanism presented in Section 2. The cooperation involved two Tabu search methods that perform well sequentially, the Unified Tabu (Cordeau, Laporte, and Mercier 2001) and Taburoute (Gendreau, Hertz, and Laporte 1994), two simple evolutionary algorithms with Order and Edge Recombination Crossover, respectively, as well as a number of post-optimization methods (2-opt, 3-opt, or-opt, and Ejection Chains) that were used to reduce the number of vehicles and

the total traveled distance. Four simple construction algorithms were used to provide initial solutions to the population.

We applied the proposed cooperation framework to the parallel method of Le Bouthillier and Crainic (2005). In the VRPTW context, the definition of an in-pattern of length n is straightforward. The problem is defined on a network, where an arc corresponds to a possible movement between two customers or between a customer and a depot. The procedures introduced in the previous section were used to define patterns.

Fixing and prohibiting arcs in the solutions explored by the four meta-heuristics is straightforward as well. Fixing (including) arcs always leads to a non empty solution space that, at the most extreme, may reduce to a single solution that represents a very (too) long pattern. Prohibiting (excluding) patterns may lead to an empty feasible solution space when patterns are too long. To avoid both situations and strike a balance between the number of feasible solutions and the size of the constrained solution space, we limit the pattern length to 25% of the size of the problem.

The computing time allocated to the cooperative method is divided into four phases: Two phases of diversification at the beginning to broaden the search, followed by two intensification phases to focus the search around promising regions. The four phases proceed as follows:

- Phase I. Built unpromising in-patterns of frequent arcs in the average subpopulation and prohibit them in the independent meta-heuristics;
- Phase II. Prohibit arcs from frequent unpromising in-patterns from the worst subpopulation;
- Phase III. Work with the average subpopulation and fix arcs from frequent promising in-patterns;
- Phase IV. Build frequent promising in-patterns from the elite sub-population and fix the arcs for the meta-heuristic searches.

Pattern lengths are explored in decreasing length in the first two phases and in increasing length in the last two phases, by increments of 1 unit.

5 Computational Experiments

The experimentation has a dual objective. On the one hand, we aim to compare the guided cooperative search to the simple version and to the best performing methods

proposed in the literature for the VRPTW and, thus, to validate our claim that the proposed method offers competitive performance in terms of both solution quality and computational effort. On the other hand, we also aim to evaluate the impact of guiding the search toward or away from specific patterns and performing diversification and intensification to control the entropy of the population.

A different search method, Taburoute, an Unified Tabu Search algorithm, and two evolutionary algorithms (OX and ER), is run on each of the four processors. The solution warehouse, the post-optimization procedures, the pattern identification method, and the construction methods are run on another processor for a total of five processors in this study.

For Taburoute, we use the parameter settings indicated in the original paper for the VRP (Gendreau, Hertz, and Laporte 1994). Tabu tags were set to a length varying between 5 and 10 iterations, 15% of the node were evaluated in the p-neighborhood dimensions, the initial solution was selected out of 15 initial generated solutions and a solution from the solution warehouse, a penalty of 1 was used for frequently moved. The parameters for the value function presented in the initial article by Cordeau, Laporte, and Mercier (Cordeau, Laporte, and Mercier 2001) (alpha:=1; beta=1; gamma=1) were used for the Unified Tabu. Finally, an arc mutation probability of 1 percent was used on temporary copies of the parents for the crossovers used by the evolutionary algorithms.

Tests have been carried out on the standard set of test problems proposed by Solomon (1987). The set contains 56 problems of 100 customers each. We also used the extended set produced by Homberger and Gehring (1999) with 300 problem instances that vary from 200 to 1000 customers. The Solomon and extended problems are divided into six categories, named C1, C2, R1, R2, RC1, and RC2. For all problem instances, customers are distributed in a [0,100] square unit. The customers in sets C are clustered together, while those in sets R are distributed randomly. Problems in sets RC combine the two characteristics. Time windows at the depot are relatively small for problems of type 1, to allow fewer customers to be served by each route; time windows are larger for problems of type 2. The service time is of 10 units by customer for problems of type R and RC, and of 90 units for class C.

Solutions in the solution warehouse are sorted, first by the number of vehicles, second by a weighted sum, C(p), of attributes: the total time required to serve all customers, the associated total distance and total waiting time at customers, and the sum of the slack left in each time window: C(p)=W1*totalTime + W2*totalDistance + W3*totalWait + W4*totalSlack.

Parameters W1 to W4 were set to 1 in all the reported experiments. This measure combined with the number of vehicles gives us an overall idea of the solution quality (totalTime and totalDistance) and flexibility (totalWait and totalSlack). The last two

measures indicate how much slack there exists in the solution and how easily feasible neighboring solutions may be explored.

In previous research (Le Bouthillier and Crainic 2005), cooperative search was found to provide faster results of equivalent or better quality that each of its independent searches. We therefore compare only simple and guided parallel cooperative searches.

Runs of 12 min wall-clock time were performed by the cooperative meta-heuristics for each of the 100 city problems. Longer running times, equal to those reported by Homberger and Gehring (1999) were allowed for the larger problem instances. These times go up to 50 min wall-clock time for the 1000 city problem. To be able to compare to the wall-clock time of the simple cooperative search, we created virtual machines under VMware and forced their CPU clocks to emulate the machines used for the previous study (Le Bouthilllier and Crainic 2005). Using faster virtual or physical machines will only decrease the wall-clock time for the same results. A virtual cluster of five Pentium III 850MHz CPU computers with 512MB of RAM under Linux was used. Computations of distances were carried out in double precision. The implementation is machine independent and can be run with MPICH on Unix, Windows, or Linux.

The four global phases that prohibit or fix arcs were each allocated 1/4 of the total wall-clock execution time. In-pattern lengths were at most 25% of the problem size.

Table 1 displays the average results per class for the cumulative number of vehicles (CNV) and distance (CTD) for the standard Solomon problems. Results of the simple Cooperative Search (**LC03**) and Guided cooperative Search (**LCK05**) are compared to those of the best methods (published or not) for the VRPTW on the benchmark site of sintef: The unified Tabu Search of Cordeau, Laporte and Mercier (2001, denoted **CLM**), the evolutionary algorithm of Homberger and Gehring (1999, denoted **HG**), the two stage hybrid local search of Bent and Van Hentenryck (2001, denoted **BVH**), the multi-start algorithm of Bräys (2004, denoted **BA**),the Active Guided Evolution Strategies of of Mester and Bräys (2004, denoted **AGES**), Rochat and Taillard (1995, denoted **RT**), Taillard *et al.* (1995, denoted **TB**), Rousseau, Gendreau, and Pesant (2002, denoted **RGP**), Bräysy, Hasle, and Dullaert (2003, denoted **BHD**), Homberger (1999, denoted **H99**), Bräysy and Dullaert (2002, denoted **EA2**), Gehring and Homberger (2001, denoted **GH01**), and Bräysy (2003, denoted **B01**).

Column LCK05 corresponds to the guided cooperative search and LC03 for the simple cooperative search. We report the best results as of February, 15 2005 on the benchmark website http://www.sintef.no/static/am/opti/projects/top/

We observe that, compared to the simple cooperative search, the new method reduced the number of vehicles by 2 and the distance by 52.38 units. We found 24 of the best known results and report the best known CNV. The guided cooperative search reports

a new best average for the R1 problem class. For C1 and C2 problem classes of 100 customers, almost all methods found the best known average number of vehicles and the total distance. For the other classes of problems, we found the best known number of vehicles in all instances and at most 0.16% of increase in distance when compared to the other methods.

Table 2 displays the results for the cumulative number of vehicles (CNV) and distance (CTD) aggregated by problem size for the extended set of problems. There are not many authors that addressed the entire problem set, which explains the limited number of entries in the table (the best methods were selected).

We observe that we obtain a new best known CTD value for problems of size 200 and 1000. For the other problem classes, the difference to the other methods is less than 0.48% in terms of the cumulative number of vehicles. In all cases, we improved upon the simple cooperative search. Since we report a slightly higher total number of vehicles compared to **HG**, we also report a reduction in the total cumulative distance by 3.94% for all classes.

6 Conclusions and Perspectives

We presented an enhanced cooperative search mechanism that creates new information from exchanged solutions and thus performs global intensification and diversification phases. The proposed framework does not suppose any specific problem structure, even though we illustrated the methodology in the context of problems defined by the inclusion/exclusion of arcs in particular networks. We also applied this methodology to the VRPTW and report very good solutions on the extended benchmark problem sets: 139 best known results found and the best known average on the problems of size 200 and 1000.

Experimental results showed that pattern identification method yields good information to guide the global search. Patterns of attributes may be constructed independently of particular solution structures and applied to a wide range of combinatorial problems. Patterns of attributes could then be used in various sequential and parallel methods to orient the search.

Cooperative search is thus quite simple to implement and the pattern identification method provides an easy mechanism to constrain the search within promising regions of the search space or away from unpromising regions. Very good quality solutions were found in linear speed up by combining good off-the-shelf methods without any particular parameter tuning. The quality of the individual methods influences the global search quality. Improved solutions may be found, however, by generating new information from

Author	R1	R2	C1	C2	RC1	RC2	CNV/CTD	Experiment
RT	12.25	2.91	10.00	3.00	11.88	3.38	415	SG. 100MHz
	1 208.50	961.72	828.38	589.86	1 377.39	1 117.44	57 231	1 run, 92.2 Min
HG	12.42	2.82	10.00	3.00	11.88	3.25	415	4 P200MHz,
	1 198	947	829	590	1 356	1 144	56 946	1 run, 5 Min
TB	12.17	2.82	10.00	3.00	11.50	3.38	410	Sun 10
	1 209.27	980.27	828.385	589.86	1 388.24	1 140.42	57 521	1 run, 248 Min
LC02	12.17	2.82	10.00	3.00	11.50	3.25	409	5 P500Mhz
	1 209.27	965.91	828.38	589.86	1 389.22	1 143.70	57 573	1 run, 12 Min
RGP	12.00	2.82	10.00	3.00	11.50	3.38	408	Sun U10
	1 203.38	950.77	828.38	589.86	1 366.45	1 093.46	56 752	n/a
CLM	12.08	2.73	10.00	3.00	11.50	3.25	407	n/a
	1 210.14	969.57	828.38	589.86	1 389.78	1 134.52	57 555	Sun U2 300MHz
LC03	12.08	2.73	10.00	3.00	**11.50**	3.25	407	5xP850 MHz,
	1 209.19	963.62	828.38	589.86	1 389.22	1143.70	57412	1 run, 12 Min
LCK05	**11.92**	2.73	10.00	3.00	**11.50**	3.25	405	5xP850 MHz,
	1 214.20	954.32	828.38	589.86	1 385.30	1129.43	57360.09	1 run, 12 Min
BHD	12.00	2.73	10.00	3.00	11.50	3.25	407	AMD 700 MHz,
	1 239.77	1 016.86	832.06	590.68	1 417.79	1 199.95	59 210	1 run, 2.6 Min
H99	11.92	2.73	10.00	3.00	11.63	3.25	406	P200 MHz,
	1 228.06	969.95	828.38	589.86	1 392.57	1 144.43	57 876	10 runs, 13 Min
EA2	12.00	2.73	10.00	3.00	11.50	3.25	406	AMD 700Mhz
	1 220.14	977.57	828.38	589.86	1 397.44	1 140.06	57 870	3 runs, 9.1 Mins
GH01	12.00	2.73	10.00	3.00	11.50	3.25	406	4 P400 MHz,
	1 217.57	961.29	828.63	590.33	1 395.13	1 139.37	57 641	5 runs, 13.5 Min
B01	11.92	2.73	10.00	3.00	11.50	3.25	405	P200 MHz,
	1 222.12	975.12	828.38	589.86	1 389.58	1 128.39	57 710	1 run, 82.5 Min
BVH	12.18	2.73	10.00	3.00	11.50	3.25	405	Sun U10 440Mhz
	1 231,08	954.18	828.38	589.86	1 384.17	1 124,47	57 272	5 run, 120 Min

Table 1: Comparison of average results on 100-customer problems

Problem	AGES		HG		BA		LC03		LCK05	
	CNV	CTD	CNV	CTD	CNV	CTD	CNV	CTD	CNV	CTD
200 total	694	168573	694	173312,321	695	170524,94	694	173061,631	**694**	**169958,49**
400 total	1389	390386	1388	409763,38	1393	399129,62	1390	410329,98	1389	396611,15
600 total	2082	796172	2076	851680,33	2084	820371,89	2088	840582,74	2086	809493,46
800 total	2765	1361586	2755	1479801,56	2776	1384306,14	2766,00	1475435,14	2761	1443399,01
1000 total	3446	2078110	3439	2236582,89	3465	2133375,35	3451	2225366,13	**3442**	**2133644,57**
Total	10376	4794827	10352	5151140,481	10413	4907707,94	10389	5124775,621	10372	4953106,68

Table 2: CNV/CTD for 200-1000 customers

the frequency of pattern appearance in the best solutions visited and by using the new information to guide the global search.

Acknowledgements

We are grateful to Jean-François Cordeau for the generous offer of sharing his Unified Tabu search code and to Michel Gendreau for his ideas and assistance with the original version of Taburoute. This research has been partially supported by the Natural Sciences and Engineering Council of Canada through its Discovery Grants program.

References

Aiex, R.M., Martins, S.L., Ribeiro, C.C., and Rodriguez, N.R. (1998). Cooperative Multi-Thread Parallel Tabu Search with an Application to Circuit Partitioning. In *Proceedings of IRREGULAR'98 - 5th International Symposium on Solving Irregularly Structured Problems in Parallel*, volume 1457 of *Lecture Notes in Computer Science*, pages 310–331. Springer-Verlag.

Bent, R. and Van Hentenryck, P. (2001). A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. *Transportation Science*. to appear.

Bräysy, O. (2001). A Reactive Variable Neighborhood Search Algorithm for the Vehicle Routing Problem with Time Windows. *INFORMS Journal on Computing*. to appear.

Bräysy, O., Hasle, G., and Dullaert, W. (2004). A Multi-Start Local Search Algorithm for the Vehicle Routing Problem with Time Windows. *European Journal of Operational Research*, 159:586–605.

Bräysy, O. and Dullaert, W. (2003). A Fast Evolutionary Metaheuristic for the Vehicle Routing Problem With Time Windows. *International Journal on Artificial Intelligence*, 12(2):153–172.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M., and Soumis, F. (2002). The VRP with Time Windows. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, SIAM, Monographs on Discrete Mathematics and Applications, chapter 7, pages 157–193. SIAM, Philadelphia, PA.

Cordeau, J.-F., Laporte, G., and Mercier, A. (2001). A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. *Journal of the Operational Research Society*, 52:928–936.

Crainic, T.G. and Gendreau, M. (2002). Cooperative Parallel Tabu Search for Capacitated Network Design. *Journal of Heuristics*, 8(6):601–627.

Crainic, T.G. and Toulouse, M. (2003). Parallel Strategies for Meta-heuristics. In F. Glover and G. Kochenberger, editors, *Handbook in Metaheuristics*, pages 475–513. Kluwer Academic Publishers, Norwell, MA.

Crainic, T.G., Toulouse, M., and Gendreau, M. (1995). Parallel Asynchronous Tabu Search for Multicommodity Location-Allocation with Balancing Requirements. *Annals of Operations Research*, 63:277–299.

Gehring, H. and Homberger, J (2001). A parallel two-phase metaheuristic for routing problems with time windows. *Asia-Pacific Journal of Operational Research*, 18(1):35–47.

Gendreau, M., Hertz, A., and Laporte, G. (1994). A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, 40:1276–1290.

Homberger, J. (2000). Verteilt-parallele metaheuristiken zur tourenplanung. Technical report, Gaber, Wiesbaden.

Homberger, J. and Gehring, H. (1999). Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows. *INFOR*, 37:297–318.

Le Bouthillier, A. and Crainic, T.G. (2005). A Cooperative Parallel Meta-Heuristic for the Vehicle Routing Problem with Time Windows. *Computers & Operations Research*, 32(7):1685–1708.

Mester, D. and Bräysy, O. (2002). Guided Evolution Strategies for Large Scale Vehicle Routing Problem with Time Windows. Working paper, Institute of Evolution, Israel University of Haifa, Israel.

Rochat, Y. and Taillard, É.D. (1995). Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*, 1(1):147–167.

Rousseau, L.-M., Gendreau, M., and Pesant, G. (2002). Using Constraint-based Operators with Variable Neighborhood Search to Solve the Vehicle Routing Problem with Time Windows. *Journal of Heuristics*, 8(1):43–58.

Solomon, M.M. (1987). Time Window Constrained Routing and Scheduling Problems. *Operations Research*, 35:254–265.

Taillard, É.D., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*, 31(2):170–186.

Annex

The Annex contains detailed results obtained by the Guided Cooperative Search metaheristic, denoted **LCK05**. Table 1 displays the results on the standard 100-customer problem sets (the only set addressed by all). The table also presents the CNV, the CTD for each method, and the experiment settings (CPU time, number of runs and minutes for each run). Table 3 displays detailed results for the standard 100-customer problem instances. Tables 4 display detailed results for the extended set of problems, from 200 to 1000 customers.

#	R1		R2		C1		C2		RC1		RC2	
01	19,00	1650,79	4,00	1252,37	10,00	828,94	3,00	591,56	14,00	1696,94	4,00	1406,94
02	17,00	1486,12	3,00	1197,66	10,00	828,94	3,00	591,56	12,00	1554,75	3,00	1389,57
03	13,00	1292,67	3,00	942,64	10,00	828,06	3,00	591,17	11,00	1262,98	3,00	1060,45
04	9,00	1007,31	2,00	825,51	10,00	824,78	3,00	590,60	10,00	1135,48	3,00	798,46
05	14,00	1377,11	3,00	994,42	10,00	828,94	3,00	588,88	13,00	1637,15	4,00	1326,83
06	12,00	1253,23	3,00	932,47	10,00	828,94	3,00	588,49	11,00	1424,73	3,00	1158,81
07	10,00	1113,69	2,00	837,20	10,00	828,94	3,00	588,29	11,00	1230,54	3,00	1062,05
08	9,00	964,38	2,00	731,23	10,00	828,94	3,00	588,32	10,00	1139,82	3,00	832,36
09	11,00	1199,63	3,00	910,55	10,00	828,94						
10	10,00	1125,04	3,00	963,37								
11	10,00	1096,73	2,00	910,09								
12	9,00	1003,73										

Table 3: Detail results LCk04, 100 customers

#	R1		R2		C1		C2		RC1		RC2	
200-01	20	4821,9	4	4502,17	20	2704,57	6	1931,44	18	3691,99	6	3136,85
200-02	18	4054,44	4	3703,86	18	2972,24	6	1863,16	18	3298,68	5	2827,45
200-03	18	3164,41	4	2976,64	18	2775,40	6	1775,11	18	3025,90	4	2617,90
200-04	18	3067,93	4	2034,20	18	2644,61	6	1720,09	18	2879,40	4	2084,89
200-05	18	4112,88	4	3367,55	20	2702,05	6	1879,01	18	3419,81	4	2912,57
200-06	18	3659,9	4	2959,14	20	2701,04	6	1857,75	18	3393,09	4	3086,76
200-07	18	3151,42	4	2532,95	20	2701,04	6	1849,46	18	3386,56	4	2550,56
200-08	18	2999,35	4	1849,87	19	2778,82	6	1824,43	18	3277,21	4	2317,80
200-09	18	3784,33	4	3115,60	18	2739,33	6	1832,04	18	3148,18	4	2291,43
200-10	18	3334,02	4	2657,00	18	2649,26	6	1806,60	18	3038,85	4	2015,60
400-01	40	10709,18	8	9403,89	40	7152,06	12	4116,14	36	8960,82	11	7019,89
400-02	36	9266,18	8	7811,05	37	7357,45	12	4048,07	36	8587,76	10	5945,68
400-03	36	7941,53	8	5988,02	36	7151,17	12	3739,72	36	7737,99	8	5298,85
400-04	36	7332,93	8	4331,07	36	6953,08	12	3535,99	36	7411,02	8	3780,66
400-05	36	9512,25	8	7143,55	40	7152,06	12	3941,05	36	8753,52	9	5970,88
400-06	36	8881,72	8	6383,54	40	7153,46	12	3877,65	36	8577,35	8	5975,26
400-07	36	7710,41	8	5378,69	39	7668,33	12	3910,61	36	8867,02	8	5727,01
400-08	36	7466,31	8	4068,97	38	7211,43	12	3803,17	36	8148,61	8	5093,18
400-09	36	8878,19	8	6493,13	37	7524,32	12	3924,39	36	7917,68	8	4745,35
400-10	36	8380,96	8	6018,84	36	6907,26	12	3729,84	36	7716,32	8	4418,64
600-01	59	21875,04	11	18833,86	60	14095,64	18	7776,16	56	17454,39	15	13275,93
600-02	55	20131,27	11	15412,26	56	14332,05	18	7512,35	55	16208,24	12	12071,40
600-03	55	17400,60	12	11915,86	56	13898,99	18	8189,67	56	15524,33	11	10466,72
600-04	55	16656,90	11	8525,42	56	13845,64	18	7216,45	55	15376,27	11	7632,79
600-05	54	20395,00	12	15854,63	60	14104,35	18	7576,35	55	17899,56	13	12324,05
600-06	54	18620,26	11	13164,53	60	14093,83	18	7500,99	55	17248,87	12	10700,42
600-07	54	18570,23	11	10697,64	60	14295,01	18	7667,39	55	17188,56	11	11687,04
600-08	54	15725,86	11	8023,64	57	14976,88	18	7483,18	55	16777,26	11	10474,95
600-09	54	19372,96	11	14379,54	56	13977,47	18	7380,21	55	16724,39	11	10346,11
600-10	54	18235,57	11	13086,16	56	14048,18	17	7523,34	55	16030,86	11	9709,96
800-01	80	37666,58	15	29383,35	80	25184,38	24	11677,72	73	31590,23	20	20869,21
800-02	72	33548,54	15	23942,54	75	25518,17	24	11422,34	72	39696,20	17	18099,68
800-03	72	31693,32	15	17992,25	72	25041,87	24	11554,18	72	35577,87	15	15711,84
800-04	72	26838,04	15	14029,92	72	24040,47	24	10963,49	72	32654,10	15	11392,25
800-05	72	34741,53	15	25837,8	80	25167,67	24	11559,98	73	31035,67	16	19639,76
800-06	72	32412,67	15	20697,06	80	25167,00	24	11418,93	73	29674,68	15	19744,73
800-07	72	30041,23	15	17058,3	80	25514,25	24	11397,54	72	43829,43	15	17964,11
800-08	72	28342,64	15	13053,31	77	25737,39	25	11410,32	72	43694,60	15	16939,31
800-09	72	34976,41	15	23734,84	74	25216,66	24	11249	72	41816,70	15	16767,66
800-10	72	32643,85	15	21929,44	72	29536,81	23	11284,46	72	41182,44	15	14892,29
1000-01	100	54724,64	19	43975,28	100	42478,95	30	16889,95	90	47143,90	22	31457,37
1000-02	92	56362,29	19	35901,41	92	42920,70	29	18312,76	90	46394,54	19	26956,04
1000-03	91	47466,88	19	25689,62	90	40934,87	29	17807,78	90	44149,27	18	20588,38
1000-04	91	43461,84	19	19551,15	90	40410,58	29	17153,19	90	41917,14	18	17449,25
1000-05	91	70838,01	19	37265,32	100	42549,13	30	16586,46	90	48853,78	18	29352,08
1000-06	91	50438,84	19	30725,20	100	42479,15	30	16473,14	90	48325,30	18	28797,14
1000-07	91	45847,84	19	25261,34	100	42689,28	31	17662,34	90	47444,73	18	27202,21
1000-08	91	42767,77	19	19294,61	97	43302,82	29	17219,59	90	45585,08	18	26037,65
1000-09	91	55275,32	19	35146,70	92	45372,21	30	16984,60	90	46325,38	18	23582,89
1000-10	91	51288,75	19	31599,84	90	40894,38	29	16555,24	90	45041,64	18	22481,03

Table 4: Detail results LCK05, 200-1000 customers